



CSE5P: Introduction to Programming (in Python)

Assignment 3

Instructor: Peter Alvaro

Reza Nasirigerdeh

Winter 2019

Instructions

1. The aim of this assignment is to practice writing modularized programs using functions in Python.
2. The deadline for the assignment is **02/01/2019 (Friday) 11:59 PM. No late submissions accepted.**
3. The grade of the assignment is out of **100 pts.**
4. Students must use Python3 (NOT Python2) to write the code.
5. Students are expected to write the code that is easily readable. To this end, they must employ meaningful variable names and comment their code properly.
6. The output of the code requires to be **precisely** as shown in the sample runs.
7. The names of the functions that students are going to define in their code must be the same as the names specified in the exercise parts (1)-(6).
8. Students must put the code in a file named `FirstName.LastName_StudentID_HW3.py` and submit it through Canvas.

Please notice that failing to comply any of these requirements will result in losing points at the discretion of the grader.

Exercise

Write a Python program that reads in the academic achievements of three international applicants and determines which one is the best candidate for the university graduate program. To this end:

1. Write a function named ***getInputs*** that reads in the academic achievements of an applicant, which are ranking of the undergraduate university (low-ranked, medium-ranked, high-ranked and of type *string*), undergraduate GPA ($[0.0-4.0]$ and of type *float*), TOEFL score ($[0-120]$ and of type *int*), GRE score ($[260-340]$ and of type *int*), and number of publications (of type *int*). The function takes no arguments and returns a tuple containing the five inputs gotten from the user [15 pts].
2. Write a function named ***computeNormGPA*** that takes the undergraduate university ranking and GPA of an applicant as arguments and returns a normalized GPA in $[0.0, 1.0]$. The normalized GPA is $(rank_weight * GPA) / 4$, in which *rank_weight* is 0.9, 1.0, 1.1 for low-ranked, medium-ranked, high-ranked, respectively. If the result of $((rank_weight * GPA) / 4)$ is greater than 1, it must be rounded to 1.0 [15 pts].
3. Write a function named ***computeNormTestScore*** that takes TOEFL score and GRE score as arguments and returns the normalized test score. The normalized test score is $(0.6 * (TOEFL_Score / 120) + 0.4 * (GRE_Score / 340))$ [15 pts].
4. Write a function named ***computeNormPubScore*** that takes the number of publications of an applicant as argument and returns normalized publication score, which is 0 if the applicant has no publication, 0.5 if the applicant has only one publication, and 1.0 if the applicant has 2 or more publications [15 pts].

5. Write a function named ***computeNormTotalScore*** that takes undergraduate university ranking, undergraduate GPA, TOEFL score, GRE score, and number of publications of an applicant as arguments and returns a normalized total score, which is $(0.5 * \text{normalized_GPA} + 0.3 * \text{normalized_test_score} + 0.2 * \text{normalized_publication_score})$ [15 pts].

Hint: Call *computeNormGPA*, *computeNormTestScore*, and *computeNormPubScore* inside this function.

6. Write a function named ***findBestCandidate*** that gets academic achievements of three applicants from the user, computes the normalized total score for each applicant, prints the total score of each applicant with four digits of precision, and determines the best candidate based on the normalized total score. The function takes no arguments and returns no value [25 pts].

Hint: Call *getInputs* to get the academic achievements of each applicant from the user and *computeNormTotalScore* to compute the normalized total score of each applicant inside this function.

Sample Run:

```
Getting the academic achievements of the first applicant ...
Enter Undergraduate university ranking (low-ranked, medium-ranked, high-ranked): medium-ranked
Enter undergraduate GPA (0.0 - 4.0): 3.8
Enter TOEFL score (0 - 120): 103
Enter GRE score (260 - 340): 307
Enter the number of publications: 1
```

```
Getting the academic achievements of the second applicant ...
Enter Undergraduate university ranking (low-ranked, medium-ranked, high-ranked): high-ranked
Enter undergraduate GPA (0.0 - 4.0): 3.4
Enter TOEFL score (0 - 120): 95
Enter GRE score (260 - 340): 301
Enter the number of publications: 0
```

```
Getting the academic achievements of the third applicant ...
Enter Undergraduate university ranking (low-ranked, medium-ranked, high-ranked): low-ranked
Enter undergraduate GPA (0.0 - 4.0): 4.0
Enter TOEFL score (0 - 120): 113
Enter GRE score (260 - 340): 320
Enter the number of publications: 2
```

```
Normalized total score of the first applicant is: 0.8379
Normalized total score of the second applicant is: 0.7162
Normalized total score of the third applicant is: 0.9324
```

```
The third applicant is the best candidate.
```